

with an on-line retailer or financial institution. As will be apparent to one of ordinary skill in the art, FIG. 1 represents one of many equivalent arrangements known in the art.

[00028] Referring now to Fig. 2, there is illustrated a block diagram of an exemplary client workstation 110. As shown, the workstation 110 is divided between internal and external components. The internal components include a Basic Input/Output System (BIOS) 13 and a processor (CPU) 12 that control the overall functioning of the computer system 10. The BIOS 13 that supports a configuration wherein floppy drives having a capacity of 720k to 1.44 MB are assigned drive letters a: and b:. The BIOS 13 also provides for fixed disks which are assigned drive letters c:, d: and so on, and are configured with a cylinder, head and sector number that relates to the capacity of the drive. A memory 14, a hard disk drive 28, a tape drive 15, a CD-ROM/R/W drive 17, and a removable media drive (e.g., floppy drive) 11 may also be connected to the CPU 12.

[00029] A removable media controller 27 serves as an interface between the removable media drive 11 and the CPU 12. For example, the removable disk controller 27 may comprise a Small Computer System Interface (SCSI) or Integrated Drive Electronics (IDE) interface controller for high capacity removable media or a floppy disk controller for conventional floppy disk drives. A hard disk controller (e.g., IDE controller) 25 serves as an interface between the CPU 12 and the hard disk 28 and the CD-ROM/R/W drive 17, and tape drive 15, respectively.

[00030] Other controllers are connected to the CPU 12 to provide an interface between a variety of external devices and the CPU 12. For example, a USB and/or parallel port controller 16, a monitor controller (video card) 18, and a keyboard and mouse controller 20 each provide an interface between CPU 12 and external parallel and/or USB devices, monitor 22, and keyboard and mouse device 24, respectively.

[00031] Typically, the workstation 110 employs the Windows® operating systems (available from Microsoft Corp., Redmond, Washington), however, other operating systems may be used. The web browser 112 may be stored on hard drive 28 and loaded by the operating system into memory 14 when operated by the web user. The client workstation 110 of Fig. 2 is presented herein for exemplary purposes only, and is not intended to limit the scope of the present invention as defined in the claims. In particular, the client workstation 110 may alternatively comprise an "Internet appliance," thin-client, or other device capable of receiving data via a browser-like application that presents the data to the user.

[00032] The web server data 160 sent by the web server 120 to the web browser 110 at sequence step 6 (FIG. 1) may include HTML code such as that illustrated and detailed in FIGS. 3A, 3B and 4 in order to prevent the inadvertent entry or submission of data to the vendor software application 142.

[00033] A more detailed description of the method of the present invention will now be described. The output data 160 may include Cascading Style Sheets (CSS) as part of a web page that also includes source code or a script (e.g., JavaScript) that may be executed by the browser 112. A “membrane” style may be defined in the output data 160 at a z-index higher than the other layers of the web page (Step 200). For example, the z-index of the membrane is set to 99 such that it is higher than other sheets in the page. In addition, the membrane is initially hidden and is positioned and sized such that it covers the entire web page to be displayed by the browser 112. A division of the Web page is associated with the “membrane” style. If z-indexing is not being used or not supported by the browser, the membrane is initially defined as hidden, without a z-index. When the user clicks a submit button, or the like to initiate a transaction with the vendor application software 142 via a web page previously downloaded as server output data 160, a submitIt function is called (Step 202). The submitIt function sends the data as input data 150 to the web sever 120 and calls a showMembrane function (Step 204).

[00034] The showMembrane function changes the Membrane style to visible (Step 206). Once the showMembrane function is called, the lower layers of a the page are inaccessible due to the higher z-index and visibility attribute of the web page division (layer) associated with the “membrane” style. This has the effect of rendering the lower layers containing data entry fields and the submit button inaccessible to the user. (On pages where z-indexing is not used, when the showMembrane function changes the “membrane” style to visible, the Web page division associated with the “membrane” is displayed, covering any existing elements). By this method, when the web page is rendered inaccessible to the user by the overlapping membrane style, subsequent entry or submissions of data (i.e., subsequent clicks of the submit button) are not possible until the page is reloaded or a new page is downloaded from the web server 120. Assuming that the “membrane” Web page division (layer) contains no text, the layer will be transparent.

[00035] The above depicts a preferred embodiment of a method in accordance with the present invention and describes the interaction and communication between a web browser and a software application and a method of preventing inadvertent entry or submission of

data. However, the present invention is not limited to an environment with a single user and a single web browser.

[00036] Various modifications of the invention, in addition to those described herein, will be apparent to those of skill in the art in view of the foregoing description. Such modifications are also intended to fall within the scope of the appended claims.